**Containers in C++ in common**

*Yasir Uddin Ahamed (Nabil)*

***Iterator returns functions and operators:***

Container\_type <class/data type> var;

Container\_type <class/data type> :: iterator P;

Container\_type <class/data type> :: reverse\_iterator RP;

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | return type | How to code | Works for |
| begin() | A iterator | P = var.begin(); | Array,deque,list, map, multimap,set,  multiset,vector,string. |
| end() | A iterator | P = var.end() – 1; | Array, deque, list, map, multimap,set,  multiset,vector,string. |
| rbegin() | A reverse iterator | RP = var.rbegin(); //last element | Array, deque, list, map, multimap, set, multiset, vector. |
| rend() | A reverse iterator | RP = var.end();  //first element | Array, deque, list, map, multimap, set, multiset, vector. |

***Capacity:***

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Return type | How to code | Works for |
| size() | Size of container(int) | int x = var.size(); | Array, deque, list, map,multimap,queue, set,multiset,stack,vector. |
| max\_size() | Maximum size (int) | int x=var.max\_size(); | Array, deque, list, map,multimap,queue, set,multiset,stack,vector. |
| resize() | change the size | var.resize(newsize,val); | deque, vector,list. |
| empty() | Empty or not (bool) | bool x = var.empty(); | Array, deque, list, map,multimap,queue, set,multiset,stack,vector. |
| shrink\_to\_fit() | reducecapacity to fitsize | var.shrink\_to\_fit(); | deque, vector. |
| capacity() | runing allocated memory | int x = var.capacity(); | vector. |

***Element Access:*** Return their own class type. Here x is same type of var, num is the number of element.

|  |  |  |  |
| --- | --- | --- | --- |
| Function/Operator | What they do | How to code | Works for |
| Operator [] | Access element | x = var[i]; (i is a int) | Array,deque,map,vector. |
| at() | Access element | x = var.at(i); (i is int) | Array,deque,map,vector. |
| front() | Access first element | x = var.front(); | Array,deque,list,vector. |
| back() | Access last element | x = var.front(); | Array,deque,list,vector. |

***Modifiers:***

|  |  |  |  |
| --- | --- | --- | --- |
| Functions Name | What they do | How to code | Works for |
| fill() | fill container with value | var.fill(x); | Array |
| swap() | swap two container | var1.swap(var2); | Array,vector,deque,list, map,multimap,queue,set, multiset, stack. |
| assign() | Assaign container content | var.assign(num,x)  var.assign(P.begin(),P.end()-1);  var.assaign(array,array+size); | deque,list,vector. |
| push\_back() | Add element at the end | var.push\_back(x); | deque,list,vector. |
| pop\_back() | Delete last element | var.pop\_back(); | deque,list,vector. |
| push\_front() | Insert element at beginning | var.push\_front(x); | deque, list. |
| pop\_front() | Delete first element | var.pop\_front(); | deque, list. |
| insert() | Insert elements | var.insert(P,x); var.insert(P,arr,arr+size);  var.insert(P,num,x);  var.insert(P,var2.begin(),var2.end()); | deque, list,  map, multimap,  set, multiset,  vector. |
| erase() | Erase eliments | var.erase(P); //one element  var.erase(P,P+n)//n elements from p | deque, list, map, multimap, set, multiset, vector. |
| clear() | Clear contents | var.clear(); | deque, list, vector,  set, multiset,  map, multimap |
| emplace() | Construct and insert element (return iterator) | P = var.emplace(P2,x);  var.emplace(P,x); | deque,list,map, multimap,queue,set, multiset,stack,vector |
| emplace\_front() | emplace at beginning | var.emplace\_front(x); | deque, list |
| emplace\_back() | emplace at end | var.emplace\_back(x); | deque, list,vector |
| emplace\_hint() | emplace with hint | var.emplace\_hint(P,x); | map, multimap, set, multiset. |
| push() | insert element | var.push(x); | queue, stack. |
| pop() | remove top/front element | var.pop(); | queue, stack. |

emplace(), emplace\_back(), data cannot be used in vector<bool> type. It has a special class hash<vector<bool>>.